GEPCI 千兆过滤卡
用户手册

GEPCI 千兆过滤卡用户手册

资料编号　DOC-GEPCI-UM
产品版本　V1.0
资料状态　发行

版权声明

© 武汉华大纵横数字技术有限公司版权所有，并保留对本手册及本声明的最终解释权和修改权。
本手册的版权归武汉华大纵横数字技术有限公司所有。未得到武汉华大纵横数字技术有限公司的书面许可，任何人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、翻译成其它语言、将其全部或部分用于商业用途。

免责声明

本手册依据现有信息制作，其内容如有更改，恕不另行通知。武汉华大纵横数字技术有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，但武汉华大纵横数字技术有限公司不对本手册中的遗漏、不准确或错误导致的损失和损害承担责任。

手册使用说明

读者对象

本手册的读者对象为安装GEPCI千兆过滤卡的工程技术人员和采用GEPCI千兆过滤卡的软件开发人员。

内容介绍

本手册详细介绍了 GEPCI 千兆过滤卡的安装方法，调试命令以及编程接口。

《GEPCI 千兆过滤卡用户手册》共分为六章：
第 1 章 产品综述 详细阐述了 GEPCI 千兆过滤卡的特性和产品规格。
第 2 章 产品硬件结构 详细阐述了GEPCI千兆过滤卡的硬件结构和选配部件介绍。
第 3 章 驱动安装 详细阐述了 GEPCI 千兆过滤卡在 windows 和 linux 下驱动和库的安装方法。
第 4 章 调试命令 详细阐述了 GEPCI 千兆过滤卡在使用过程中的需要调试和查看状态的命令。
第 5 章 软件编程接口 详细阐述了 GEPCI 千兆过滤卡提供的软件编程接口。
第 6 章 问与答 详细介绍了 GEPCI 千兆过滤卡在安装过程出现的一些常见问题和解决方案。

手册约定

1．通用格式约定

格 式	意 义
宋体	正文采用宋体表示。
黑体	除一级标题采用宋体**加粗**以外，其余各级标题均采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display”格式	自定义的“Terminal Display”格式（英文 Courier New； 中文 宋体； 文字大小 8.5）表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用**加粗**字体表示。

2．命令行格式约定

格 式	意 义
粗体	*命令行关键字（命令中保持不变、必须照输的部分）采用加* *粗字体表示。*
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜* *体*表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从两个或多个选项中选取一个。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... } *	表示从两个或多个选项中选取多个，最少选取一个，最多选 取所有选项。
[x | y | ...] *	表示从两个或多个选项中选取多个或者不选。

获取技术支援

客户在产品使用及网络运行过程中遇到问题时请随时与武汉华大纵横数字技术有限公司的服务支持热线联系。此外，客户还可通过武汉华大纵横数字技术有限公司网站及时了解最新产品动态，以及下载需要的技术文档。

目　录

GEPCI 千兆过滤卡 1

第 **1** 章　产品综述 **1-1**

1.1　**GEPCI** 千兆过滤卡简介 1-1
1.2　产品规格 1-2
1.3　产品订购 1-2

第 **2** 章　产品硬件结构 **2-3**

2.1　**GEPCIQ02S02S-SFT** 2-3
2.1.1　硬件结构 2-3
2.1.2　系统配置清单 2-3
2.2　**GEPCIQ04S04S-SFT** 2-4
2.2.1　硬件结构 2-4
2.2.2　系统配置清单 2-5

第 **3** 章　驱动安装 **3-6**

3.1　Windows 下驱动安装 3-6
3.1.1　驱动安装 3-6
3.1.2　动态链接库安装 3-10
3.1.3　千兆过滤卡测试 3-10
3.2　Linux 下驱动安装 3-11
3.2.1　驱动安装 3-11
3.2.2　动态链接库安装 3-11
3.2.3　千兆过滤卡测试 3-11

第 **4** 章　调试命令 **4-13**

4.1　**Logout** 4-13
4.2　**SET FILTER-RULE** 4-13
4.3　**SHOW FILTER-RULE** 4-14
4.4　**DELETE FILTER-RULE** 4-14
4.5　**DELETE ALL-FILTER** 4-14
4.6　**READ REGISTER** 4-14
4.7　**WRITE REGISTER** 4-15
4.8　**READ MEMORY** 4-15
4.9　**WRITE MEMORY** 4-15

4.10 WRITE CAM-ENTRY 4-16
4.11 READ CAM-ENTRY 4-16
4.12 CAM-SEARCH 4-16
4.13 SET PMRC RX-PARITY 4-16
4.14 SET PMRC LONG-PKT 4-17
4.15 SET PMRC SHORT-PKT 4-17
4.16 SET PMRC RX-MODE 4-17
4.17 SET PMRC VLAN 4-18
4.18 SET PMRC MPLS 4-18
4.19 START TEST SYSTEM 4-18
4.20 START TEST CAM 4-18
4.21 START TEST MEM 4-19
4.22 START TEST IXF1104 4-19
4.23 START TEST PMRC 4-19
4.24 START TEST IXF1104_LOOPBACK 4-20
4.25 SHOW IXF1104 STATISTICS 4-20
4.26 SHOW IXF1104 STATUS 4-20
4.27 SHOW IXF1104 CONFIG 4-21
4.28 SET IXF1104 PARITY 4-22
4.29 SET IXF1104 PUSHBACK 4-22
4.30 SET IXF1104 MAX-FRAME 4-22
4.31 SET IXF1104 AUTO-NEGOTIATION 4-23
4.32 SET IXF1104 PADDING 4-23
4.33 SET IXF1104 CRC-APPENDING 4-23
4.34 SHOW STATISTICS 4-23
4.35 SHOW PMRC STATISTICS 4-24
4.36 SHOW PMRC CONFIG 4-24
4.37 SET CARD 4-25

第 5 章 应用编程接口 5-26

5.1 接口概述 5-26
5.2 基本数据结构 5-26
5.2.1 pcap_t 5-26
5.2.2 pcap_handler 5-26
5.2.3 pcap_pkthdr 5-26
5.3 功能详述 5-27
5.3.1 pcap_open_live 5-27
5.3.2 pcap_compile 5-27
5.3.3 pcap_set_default_rule 5-28

5.3.4 pcap_loop *5-28*
5.3.5 pcap_breakloop *5-28*
5.3.6 pcap_close *5-29*
5.3.7 pcap_stats_gfilter *5-29*
5.3.8 pcap_version *5-29*
5.4 规则配置 5-30
5.4.1 基本规则 *5-30*
5.4.2 规则的组合运算 *5-30*
5.4.3 过滤规则 *5-31*
5.5 接口函数使用举例 5-32
5.5.1 举例一……接收 tcp port 80 数据包 *5-32*
第 **6** 章 问与答 **6-33**

第1章　产品综述

1.1　GEPCI 千兆过滤卡简介

GEPCI 千兆过滤卡主要用于对千兆以太网络中的数据包进行分析，并根据数据包协议头中的某些字段对数据包进行分流。目前，GEPCI 千兆过滤卡支持的分类规则为：目的 IP 地址＋源 IP 地址＋协议号＋目的端口号＋源端口号＋TCP 标志位。与采用普通的千兆网卡对数据包进行分析相比，GEPCI 千兆过滤卡具有以下特点：

1. GEPCI 千兆过滤卡可以同时支持 2 路或 4 路输入，并将输入的数据进行合并后输入服务器；普通的千兆网卡只能支持 1 路输入，需要由服务器将不同网卡输入的数据进行合并。对于千兆以太网络比较多，但每一路流量又不是很大的情况下，采用 GEPCI 千兆过滤卡可以有效的减少服务器数量。

2. GEPCI 千兆过滤卡能够处理的最大数据量是 250 万 pps（每秒数据包数），而普通的千兆卡能够处理的最大数据量是 35 万 pps。当千兆以太网络流量超过 35 万 pps 时，普通的千兆网卡就会丢掉数据包，这样进入服务器的数据包就会不完整。

3. GEPCI 千兆过滤卡采用硬件对数据包进行分类，并将分类好的数据送往不同的应用进程进行处理；而普通的千兆网卡，需要采用软件来对数据包进行分类，这样配置的分类规则就不能很多，而 GEPCI 过滤卡能够支持多达 16000 条分类规则。

4. GEPCI 千兆过滤卡每次传输 64K 字节的数据量产生一次中断；而普通的千兆网卡每接收到一个数据包就会产生一次中断。在 CPU 的占用率上，采用 GEPCI 千兆过滤卡接收数据只是采用普通网卡接收数据的一半。

5. GEPCI 千兆过滤卡可以作为一个网桥，支持 2 个或 4 个千兆以太网络之间数据包的线速转发。同时可以根据用户的需求，将满足特定要求的数据包丢弃或者输入到服务其中进行分析。而普通的千兆网卡不具备这一功能。

6. GEPCI 千兆过滤卡支持对网络上的数据包进行统计；目前可以对 65536 个源端口、65536 个目的端口和 256 个协议的数据包进行统计。而普通的千兆网卡不具备这一功能。

7. GEPCI 千兆过滤卡支持采样功能，当某一种类别的数据量比较大，服务器处理不了时，就可以对输入的数据进行采样，只允许一部分数据进入。而普通的网卡不具备这一功能。

8. GEPCI 千兆过滤卡由于采用 SFP 接口，可以同时支持电口和光口；而普通的网卡只能支持电口或者光口中的一种。

1.2　产品规格

为了满足不同用户对 GEPCI 千兆过滤卡的需求，我们开发出了一系列 GEPCI 千兆过滤卡，其命名规则如下：

GEPCI　X　XX　X　XX　X　—　X　X　X

	PCI 类型 Q：64bit L：32bit	输入端口数 如 02，04 等	输入端口类型 E：电口 O：光口 S：SFP	输出端口数 如 02，04 等	输出端口类型 E：电口 O：光口 S：SFP	采样功能 S：采样	转发功能 F:转发	发送功能 T：发送

例如 GEPCIQ02S02S-SFT，就表示该千兆过滤卡用于 64bitPCI 插槽；具有两个输入端口和两个输出端口，且输入端口和输出端口均为 SFP 类型；并且该过滤卡支持采样，装发和发送功能。

1.3　产品订购

产 品 名 称	描　述
GEPCIQ02S02S	64 位 GEPCI 卡，两个千兆 SFP 光口（LC）
GEPCIQ04S04S	64 位 GEPCI 卡，四个千兆 SFP 光口（LC）

第2章　产品硬件结构

下面以 GEPCIQ02S02S-SFT 和 GEPCIQ04S04S-SFT 为例，介绍 GEPCI 千兆过滤卡的硬件结构特点，该系列的其他产品与此类似。

2.1　GEPCIQ02S02S-SFT

2.1.1　硬件结构

1. GEPCIQ02S02S-SFT 正面结构如下图所示：

图 2-1 GEPCIQ02S02S-SFT 正面图

2．GEPCIQ02S02S-SFT 面板灯的说明

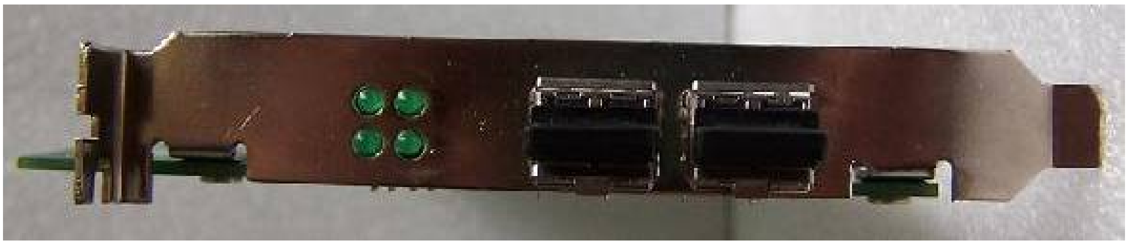

图 2-2 GEPCIQ02S02S-SFT 测面图

正面向上，光口从左至右为 0 口和 1 口，左边的面板灯对应 0 口，上面为 TX0，下面为 RX0；右边的面板灯对应 1 口，上面的 TX1，下面为 RX1。

2.1.2　系统配置清单

项　目	部件名称	数　量
1	千兆过滤卡	1 块
2	1.25G SFP 光收发器	2 个

表 2-1 基本配置清单

2.2　GEPCIQ04S04S-SFT

2.2.1　　硬件结构

1. GEPCIQ04S04S-SFT 正面结构如下图所示：

图 2-3 GEPCIQ04S04S-SFT 正面图

2．GEPCIQ04S04S-SFT 面板灯的说明：

图 2-4 GEPCIQ04S04S-SFT 测面图

正面向上，光口从左至右为 0 口、1 口、2 口和 3 口，左边下面的面板灯对应 0 口，为 RX0－TX0；右边下面的面板灯对应 1 口，为 RX1－TX1；左边上面的面

板灯对应 2 口，为 RX2－TX2；右边上面的面板灯对应 3 口，为 RX3－TX3。

2.2.2　系统配置清单

项 目	部件名称	数 量
1	千兆过滤卡	1 块
2	1.25G SFP 光收发器	4 个

表 2-2 基本配置清单

第3章　驱动安装

3.1　Windows 下驱动安装

本章介绍了 GEPCI 千兆过滤卡在 windows 下驱动和动态链接库的安装方法。安装所需的文件清单如下：GePci.sys 和 GePci.inf 是千兆卡的驱动和安装文件；wpcap.lib 和 wpcap.dll 是应用程序需要的动态链接库；include 是需要包含的头文件。

3.1.1　驱动安装

● 将千兆过滤卡插入系统中，出现如图 3-1 所示画面，点击“下一步”。

图 3-1 找到新硬件

● 出现如图 3-2 所示画面，选择“搜索适于我的设备的驱动程序”，然后点击“下一步”。

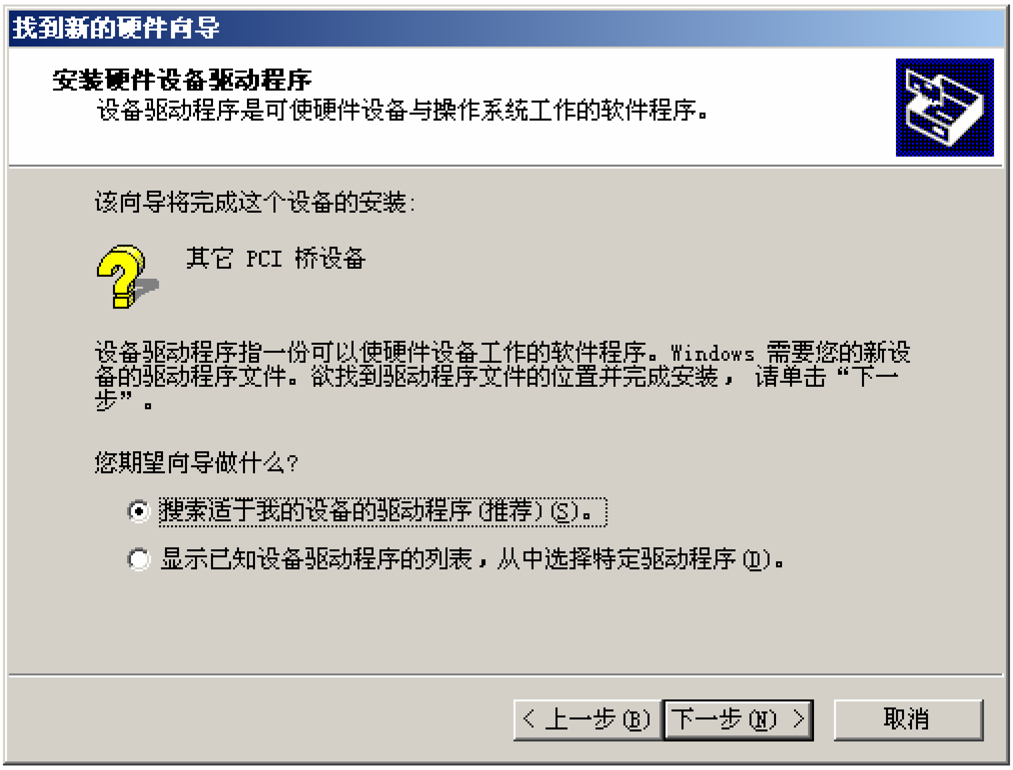

图 3-2 安装硬件设备驱动程序

- 出现如图 3-3 所示画面，选择“指定一个位置”，点击“下一步”。

图 3-3 找到驱动程序文件

- 出现如图 3-4 所示画面，点击“浏览”，选择设备驱动程序所在目录中的“GePci.inf”文件，点击“确定”。

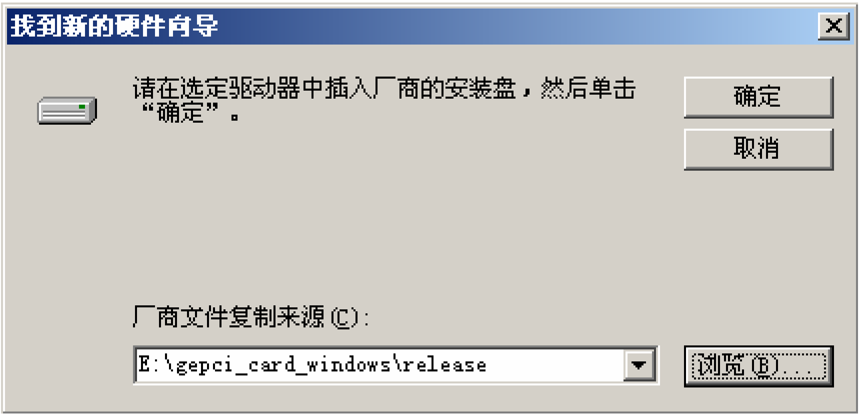

图 3-4 厂商文件复制来源

- 出现如图 3-5 所示画面，点击“下一步”。

图 3-5 查找驱动程序

- 出现如图 3-6 所示画面，点击“完成”，此时表示已将驱动程序成功安装到系统中。

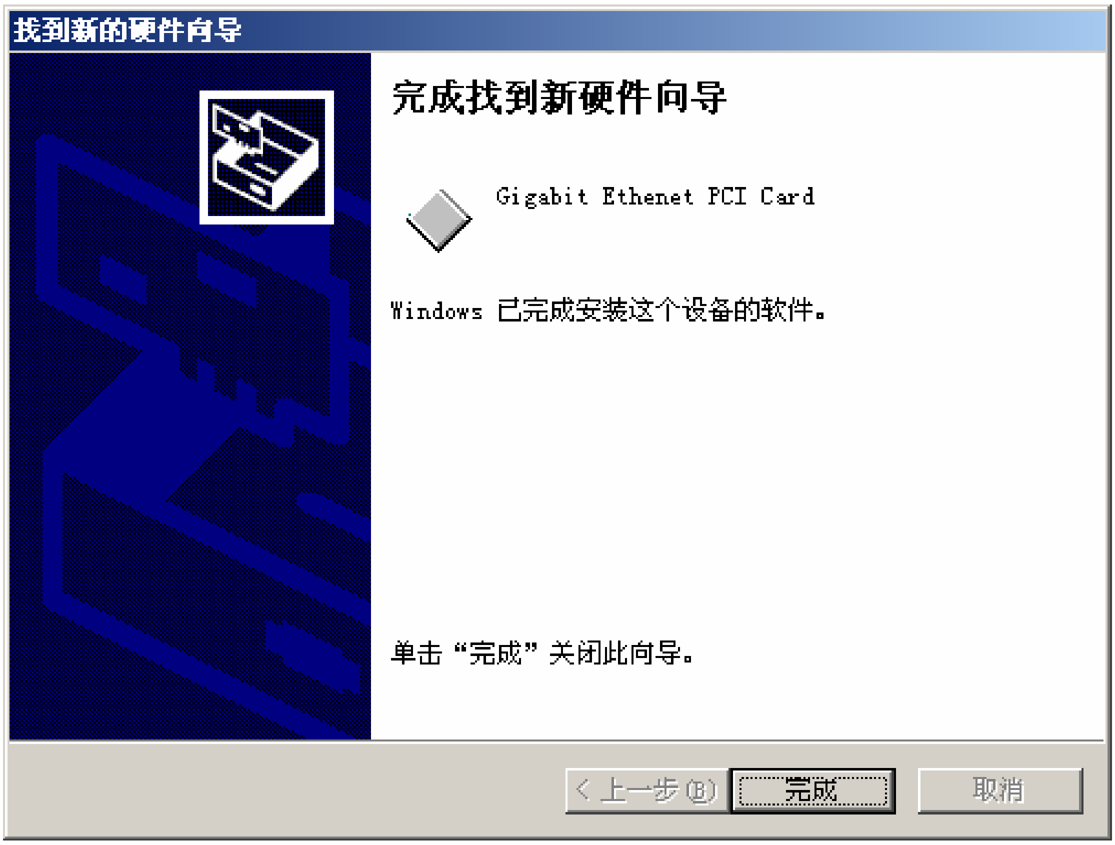

图 3-6 完成驱动程序安装

安装完成后，在"设备管理器"的"其他设备"中出现"Gigabit Ethenet PCI Card"，如图 3-7 所示

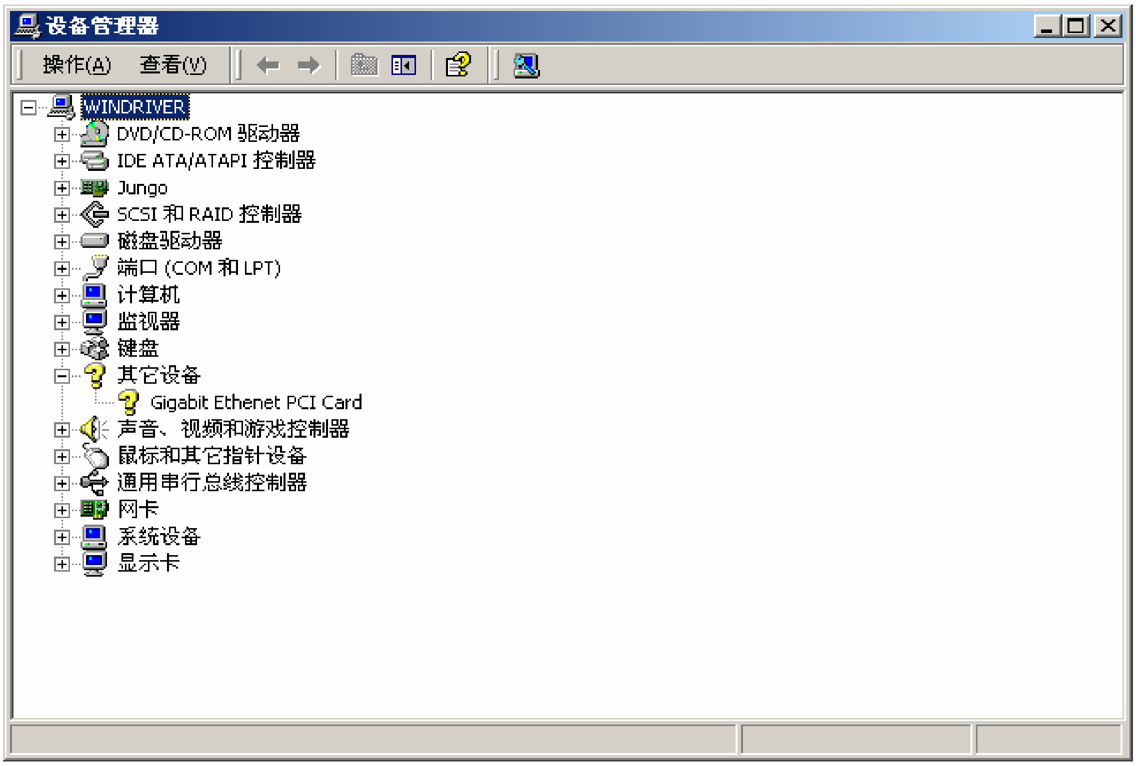

图 3-7 设备管理器

3.1.2　动态链接库安装

运行 WinPcap_3_1.exe，安装 winpcap 库，然后用附件中 wpcap.dll 替换 windows 安装目录下 system32 中的 wpcap.dll 文件。并将 wpcap.lib 添加到你们程序的工程中，并包含附件中的头文件。

3.1.3　千兆过滤卡测试

example 中有一个 test 程序，用于对安装好的 GEPCI 千兆过滤卡进行测试。用户可以使用这个测试程序通过千兆过滤卡来采包。点击 test 程序，出现“please input the rule”提示，然后直接回车，此时会采集所有数据包，并在每收到 1000 个数据包时打印一条信息，如图 3-8 所示：

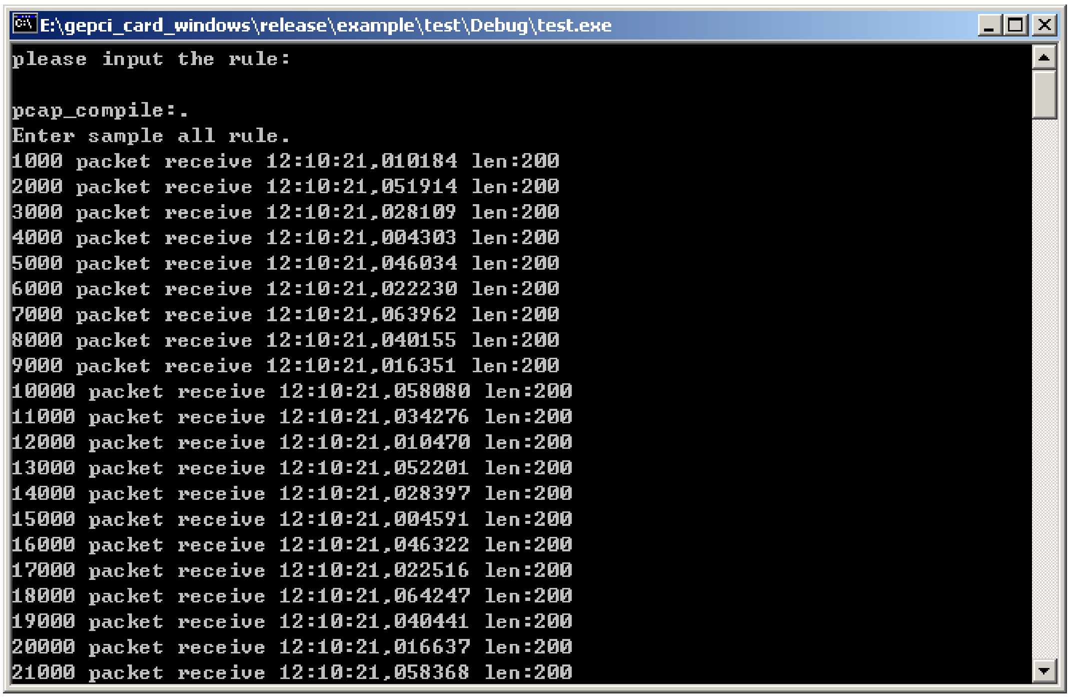

图 3-8 采集所有数据报测试程序

如果要采集端口号是 80 的数据包时，请先点击 test 程序，出现“please input the rule”提示，然后输入过滤规则，如：“port 80”，千兆过滤卡将采集卡采集所有端口号为 80 的数据包，并在每收到 1000 个数据包时打印一条信息，如图 3-9 所示：

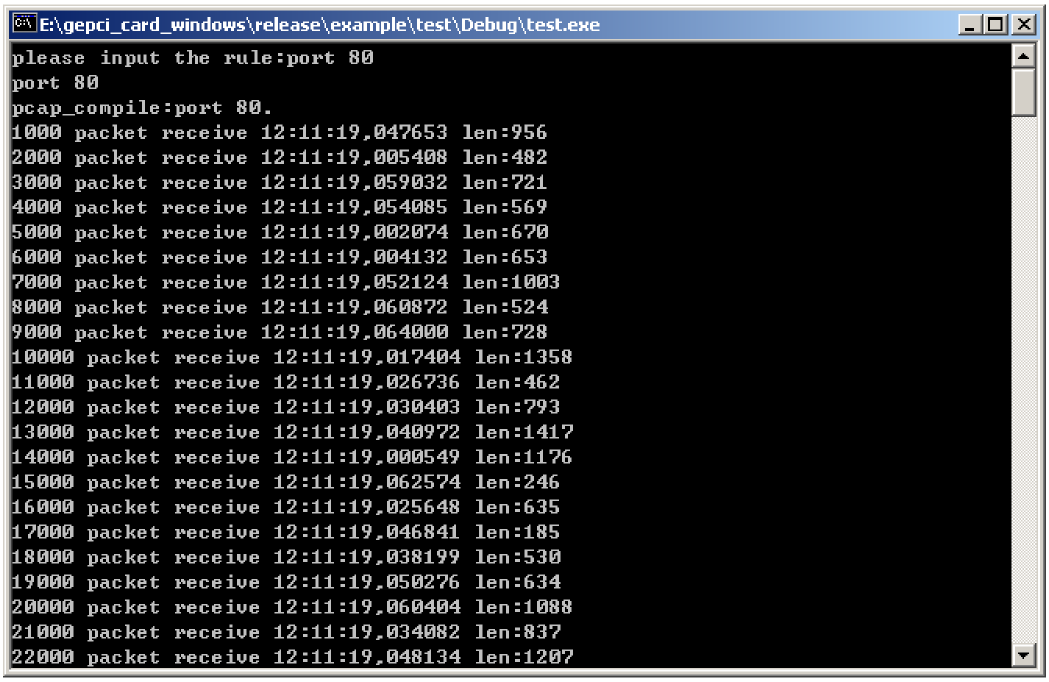

图 3-9 规则过滤测试程序

如果能正常完成图 3-8 的测试程序，表明驱动程序安装正确，否则请重新安装驱动程序。

3.2 Linux 下驱动安装

3.2.1 驱动安装

- 运行命令 tar xvzf gepci_card_linux_release.tgz 解开软件包。
- 运行命令 cd gepci_card_linux，进入解开的软件包中。
- 运行命令 cd gepci_card_driver 进入驱动安装目录。
- 运行命令./build,安装驱动。

3.2.2 动态链接库安装

- 运行命令 cd gepci_card_libpcap 进入动态链接库安装目录。
- 运行命令./build，安装 libpcap 库。

3.2.3 千兆过滤卡测试

- 运行命令 cd tcpdump-3.8.3 进入测试目录，运行./tcpdump -nne,采集所有数据包，如图 3-10 所示：

```
[fb@localhost tcpdump-3.8.3]$ ./tcpdump -nne
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on gfilter, link-type EN10MB (Ethernet), capture size 96 bytes
11:22:30.019866 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 891: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 123456:124293(837) ack 234567 win 40
96
11:22:30.019941 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 282: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:228(228) ack 1 win 4096
11:22:30.021811 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 1300: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:1246(1246) ack 1 win 4096
11:22:30.021935 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 480: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:426(426) ack 1 win 4096
11:22:30.023109 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 888: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:834(834) ack 1 win 4096
11:22:30.023182 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 276: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:222(222) ack 1 win 4096
11:22:30.025031 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 1288: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:1234(1234) ack 1 win 4096
11:22:30.025150 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 457: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:403(403) ack 1 win 4096
11:22:30.026235 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 842: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:788(788) ack 1 win 4096
11:22:30.026286 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 185: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:131(131) ack 1 win 4096
11:22:30.027796 00:00:00:00:00:1f > 00:00:00:00:00:00, ethertype IPv4 (0x0800), leng
th 1106: IP 198.19.1.2.1024 > 198.19.1.1.1025: . 0:1052(1052) ack 1 win 4096
```

图 3-10 采集所有数据报测试程序

第4章　调试命令

本章详细介绍了用于调试和查看 GEPCI 千兆过滤卡状态的命令。

4.1　Logout

该命令主要用于退出调试程序。

命令格式

logout

参数说明

无。

具体描述

logout

说明：退出调试程序。

4.2　set filter-rule

该命令用于配置千兆过滤卡的规则。

命令格式

set filter-rule [index *priority* **]　[protocol** *pt* **] [srcip** *sip* **] [dstip** *dip* **] [srcport** *sp* **] [dstport** *dp* **]　{ drop | forward } [ixf-enable** *port* **| ixf-disable] [plx-enable | plx-disable] [sample** *base ratio* **]**

参数说明

index	设置过滤规则索引号，十六进制数，范围：[0x30~0x3ffe]。
protocol	设置过滤规则的协议类型，范围：tcp / udp 。
srcip	设置过滤规则的源 IP，例如 192.168.88.1 。
dstip	设置过滤规则的目的 IP，例如 192.168.88.1 。
srcport	设置过滤规则的源端口，例如 8080。
dstport	设置过滤规则的目的端口，例如 8080。
forward	将符合过滤规则的数据包转发。
ixf-enable	将符合过滤规则的数据包转发到的网络端口,范围：[1~4]。
ixf-disable	符合过滤规则的数据包不转发到网络端口。
plx-enable	将符合过滤规则的数据包收到服务器。
plx-disable	符合过滤规则的数据包不收到服务器。
sample	数据包采样率设置。

具体描述

set filter-rule index 66 protocol dstip 192.168.88.2 srcport 1024 forward ixf-enable 1

说明：设置一条索引号为 0x66 的规则，将目的 ip 为 192.168.88.2，且源端口为 1024 的 tcp 包由网络端口 1 转发。

4.3　show filter-rule

该命令用于显示已配置的过滤规则。

命令格式

show filter-rule [index *priority***]**

参数说明

index　　要查看的规则的索引号，省略代表显示所有规则。

具体描述

show filter-rule

说明：查看所有的过滤规则。结果如图所示：

```
GEPCI>set filter-rule index 30 protocol tcp srcip 1.1.1.1 forward ixf-enable 0
GEPCI>show filter-rule
index sip     dip     sp   dp   pt  port  ixf  plx  ratio pid macMode Data Mod
48    1.1.1.1 0.0.0.0 0    0    tcp   0   YES  YES  N/A    0    0      0   1
GEPCI>_
```

4.4　delete filter-rule

该命令用于删除已配置的过滤规则。

命令格式

delete filter-rule {index *priority***}**

参数说明

index　　要删除的规则的索引号。

具体描述

delete filter-rule 66

说明：删除索引号为 0x66 的过滤规则。

4.5　delete all-filter

该命令用于删除所有已配置的过滤规则。

命令格式

delete all-filter

参数说明

无。

具体描述

delete all-filter

说明：删除所有已配置的过滤规则。

4.6　read register

该命令用于读取指定寄存器的内容。

命令格式

read register {*type* **} {***offset***}**

参数说明

type 指定访问的芯片，范围：pmrc ixf1104 plx9656。
offset 指定寄存器的偏移地址，十六进制数。

具体描述

read register pmrc 0

说明：读取芯片 pmrc 上偏移地址为 0x0 的寄存器的内容

4.7 write register

该命令用于设置指定寄存器的值。

命令格式

read register {*type* **}** **{***offset***}** **{***value***}**

参数说明

type 指定访问的芯片，范围：pmrc / ixf1104 / plx9656。
offset 指定寄存器的偏移地址，十六进制数。
value 将要写入寄存器的值，十六进制数。

具体描述

write register pmrc 0 ff

说明：将芯片 pmrc 上偏移地址为 0x0 的寄存器的内容置为 0xff。

4.8 read memory

该命令用于读取指定存储器的内容。

命令格式

read memory {*type* **}** **{***offset***}**

参数说明

type 指定访问的芯片，范围：mac / extern-mem。
offset 指定寄存器的偏移地址，十六进制数。

具体描述

read memory extern-mem 0

说明：读取 extern-mem 上偏移地址为 0x0 的寄存器的内容

4.9 write memory

该命令用于设置指定存储器的值。

命令格式

read register {*type* **}** **{***offset***}** **{***value***}**

参数说明

type 指定访问的芯片，范围：mac / extern-mem。
offset 指定寄存器的偏移地址，十六进制数。
value 将要写入寄存器的值，十六进制数。

具体描述

write memory extern-mem 0 ff

说明：将 extern-mem 上偏移地址为 0x0 的寄存器的内容置为 0xff。

4.10 write cam-entry

该命令用于设置 cam 的内容。

命令格式

write cam-entry {*address*} {*key*} {*mask*}

参数说明

address cam 的入口地址，十六进制数。
key 写入 cam 的值，96 位十六进制数。
mask 写入 cam 的掩码，96 位十六进制数。

具体描述

write cam-entry 30 11-22-33-44-55-66-77-88-99-00-11-22-33-44-55-66-77-88 ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff

说明:把 cam 的地址为 0x30 处的值设为 11-22-33-44-55-66-77-88-99-00-11-22-33-44-55-66-77-88，掩码设为 ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff-ff。

4.11 read cam-entry

该命令用于读取 cam 的内容。

命令格式

read cam-entry {*start*} {*end*}

该命令用于读取 cam 的内容。

参数说明

start 希望读取的地址段的起始地址，十六进制数。
end 希望读取的地址段的结束地址，十六进制数。

具体描述

read cam-entry 30 4f

说明：读取从地址 0x30 到 0x4f 之间的 cam 的内容。

4.12 cam-search

该命令用于获得与输入值匹配的 cam 地址。

命令格式

cam-search {mask} {key}

参数说明

mask 设置全局掩码索引，十六进制数，范围：[0~7]。
key 用于匹配的 cam 值。96 位十六进制数。

具体描述

cam-search 0 11-22-33-44-55-66-77-88-99-00-11-22-33-44-55-66-77-88

说明：在 cam 中以索引号为 0 的全局掩码，匹配 11-22-33-44-55-66-77-88-99-00-11-22-33-44-55-66-77-88，返回得到的 cam 地址。

4.13 set pmrc rx-parity

该命令用于设置 pmrc 的奇偶校验方式。

命令格式

set pmrc rx-parity {*type*}

参数说明

type　　pmrc 的奇偶校验方式，范围：odd / even。

具体描述

set pmrc rx-parity odd

说明：将 pmrc 的奇偶校验方式设为奇校验。

4.14　set pmrc long-pkt

该命令用于设置 pmrc 入口的最长包长。

命令格式

set pmrc long-pkt {*size*}

参数说明

size　　pmrc 入口的最长包，十进制数，范围：[536~65535]。

具体描述

set pmrc long-pkt 1500

说明：设置 pmrc 入口的最长包长为 1500。

4.15　set pmrc short-pkt

该命令用于设置 pmrc 入口的最短包长。

命令格式

set pmrc short-pkt {*size*}

参数说明

size　　pmrc 入口的最短包长，十进制数，范围：[1~1500]。

具体描述

set pmrc long-pkt 30

说明：设置 pmrc 入口的最长包长为 30。

4.16　set pmrc rx-mode

该命令用于设置 pmrc 入口开关。

命令格式

set pmrc rx-mode {*type*}

参数说明

type　　pmrc 入口开关值，范围：enable / disable。

具体描述

set pmrc rx-mode disable

说明：关闭 pmrc 入口。

4.17 set pmrc vlan

该命令用于设置 pmrc 的 vlan 处理模式。

命令格式

set pmrc rx-vlan {*type*}

参数说明

type pmrc 的 vlan 处理模式，范围：remove / bypass。

具体描述

set pmrc rx-vlan remove

说明：设置 pmrc 的 vlan 处理模式为 remove。

4.18 set pmrc mpls

该命令用于设置 pmrc 的 mpls 处理模式。

命令格式

set pmrc rx-mpls {*type*}

参数说明

type pmrc 的 mpls 处理模式，范围：remove / bypass。

具体描述

set pmrc rx-mpls remove

说明：设置 pmrc 的 mpls 处理模式为 remove。

4.19 start test system

该命令用于开始系统自检。

命令格式

start test system

参数说明

无

具体描述

start test system

说明：开始系统自检。

4.20 start test cam

开始自检 cam。

命令格式

start test cam {*start*} {*end*} {*type*}

参数说明

start 检测的起始地址，范围：[0~0xFFFF]。

end　　检测的终止地址，范围: [0~0xFFFF]。
type　　检测模式，范围：[0~9]。

具体描述

start test cam 0 FF 2

说明：开始自检 cam，地址从 0 到 FF，模式为 2。

4.21 start test mem

该命令用于开始自检 mem。

命令格式

start test mem {*start*} {*end*} {*type*}

参数说明

start　　检测的起始地址，十六进制数，范围：[0~0xFFFFFFFF]。
end　　检测的终止地址，十六进制数，范围：[0~0xFFFFFFFF]。
type　　检测模式，十进制数，范围：[1~9]。

具体描述

start test mem 0 FF 2

说明：开始自检 mem，地址从 0 到 FF，模式为 2。

4.22 start test ixf1104

该命令用于开始自检 ixf1104。

命令格式

start test ixf1104

参数说明

无。

具体描述

start test ixf1104。

说明：开始自检 ixf1104。

4.23 start test pmrc

该命令用于开始自检 pmpc。

命令格式

start test pmpc

参数说明

无。

具体描述

start test pmpc

说明：开始自检 pmpc。

4.24　start test ixf1104_loopback

该命令用于开始自检 ixf1104_loopback。

命令格式

start test ixf1104_loopback

参数说明

无。

具体描述

start test ixf1104_loopback

说明：开始自检 ixf1104_loopback。

4.25　show ixf1104 statistics

该命令用于查看 ixf1104 的统计信息。

命令格式

show ixf1104 statistics {*port*}

参数说明

port　　　　端口号，范围：[1~4]。

具体描述

show ixf1104 statistics 1

说明：显示 ixf1104 端口 1 的统计信息。结果如图所示：

```
GEPCI>show ixf1104 status 1
Auto negotiation:
An complete:     0
Rx Sync:         0
Rx Config:       0
Invalid Word:    1
Carrier Sense:   1
Next Page:       0
Remote Fault:    0
Asym Pause:      0
Sym Pause:       0
Half Duplex:     0
Full Duplex:     0
TX FIFO:     overflow: 0       underflow: 0       out of seqeunce: 0
SerDes signal detect: 1
GBIC:        Rx_LOS: 0    TX_FAULT: 0    MOD_DEF: 0
```

4.26　show ixf1104 status

该命令用于查看 ixf1104 的状态。

命令格式

show ixf1104 status {*port*}

参数说明

port　　　　端口号，范围：[1~4]。

具体描述

show ixf1104 status 1

说明：显示 ixf1104 端口 1 的状态。结果如图所示。

```
GEPCI>show ixf1104 statistic 1
SPI3 address parity error drop sts(70A):          0
TX FIFO:
SPI3 address parity error drop sts(70A):          0
        overflow frame drop sts(621~624):         0
        error frame drop sts(625~629):            0
        occupancy sts(62d~630):                   0
MAC tx statistic:
txOctetsOK:                            0  ¦  txOctetsBad:                        0
txUCPkts:                              0  ¦  txMCPkts:                           0
txBCPkts:                              0  ¦  txPkts64Octets:                     0
txpkts65to127Octets:                   0  ¦  txpkts128to255Octets:               0
txpkts256to511Octets:                  0  ¦  txpkts512to1023Octets:              0
txpkts1024to1518Octets:                0  ¦  txpkts1519toMaxOctets:              0
txDeferred:                            0  ¦  txTotalCollisions:                  0
txSingleCollisions:                    0  ¦  txMultipleCollisions:               0
txLateCollisions:                      0  ¦  txExcessiveCollisionErrors:         0
txExcessiveDeferralErrors:             0  ¦  txExcessiveLengthDrop:              0
txUnderrun:                            0  ¦  txTagged:                           0
txCRCError:                            0  ¦  txPauseFrames:                      0
txFlowControlCollisionsSend:           0
MAC rx statistic:
rxOctetsOK:                     1b930000  ¦  rxOctetsBad:                     1b93
rxUCPkts:                              0  ¦  rxMCPkts:                           0
rxBCPkts:                          e0000  ¦  rxPkts64Octets:                     0
rxpkts65to127Octets:                   0  ¦  rxpkts128to255Octets:               0
rxpkts256to511Octets:              e0000  ¦  rxpkts512to1023Octets:              e
rxpkts1024to1518Octets:                0  ¦  rxpkts1519toMaxOctets:              0
rxFCSErrors:                           0  ¦  rxTagged:                           0
rxDataError:                           0  ¦  rxAlignErrors:                      0
rxLongErrors:                          0  ¦  rxJabberErrors:                     0
rxPauseMacCntrl:                       0  ¦  rxUnknownMacCntrl:                  0
rxVeryLongErrors:                      0  ¦  rxRuntErrors:                       0
rxShortErrors:                         0  ¦  rxCarrierExtendError:               0
rxSequenceErrors:                      0  ¦  rxSymbolErrors:                     0
```

4.27 show ixf1104 config

该命令用于查看 ixf1104 的配置信息。

命令格式

show ixf1104 config

参数说明

无。

具体描述

show ixf1104 config

说明：显示 ixf1104 的配置信息。结果如图所示：

```
GEPCI>show ixf1104 config
IXF1104 Configuration:
SPI3 parity check type : Odd
TX FIFO trigger threshold :  80
TX FIFO high water mark :  1024
TX FIFO low  water mark :  896
Maximum frame size :  1518
Auto negotiation : Disable
Pad packet up to 64 byte : Enable
CRC appending : Enable
GEPCI>
```

4.28 set ixf1104 parity

该命令用于设置 ixf1104 的奇偶校验模式。

命令格式

set ixf1104 parity {*parity*}

参数说明

parity 奇偶校验模式，范围：odd / even。

具体描述

set ixf1104 parity odd

说明：设置 ixf1104 为奇校验模式。

4.29 set ixf1104 pushback

该命令用于设置 ixf1104 的流量控制。

命令格式

set ixf1104 pushback {*high*} {*low*}

参数说明

high 流量上限，十进制数，范围：[1~4095]。
low 流量下限，十进制数，范围：[1~4095]。

具体描述

set ixf1104 pushback 3000 500

说明：设置 ixf1104 的流量上限为 3000，下限为 500。

4.30 set ixf1104 max-frame

该命令用于设置 ixf1104 的最大帧长。

命令格式

set ixf1104 max-frame {*maxframe*}

参数说明

maxframe 最大帧长，十进制数，范围：[1~16383]。

具体描述

set ixf1104 max-frame 10000

说明：设置 ixf1104 的最大帧长为 10000。

4.31　set ixf1104 auto-negotiation

该命令用于设置 ixf1104 的自动协商开关。

命令格式

set ixf1104 auto-negotiation {*type*}

参数说明

type　　自动协商开关，范围：enable / disable。

具体描述

set ixf1104 auto-negotiation enable

说明：允许 ixf1104 自动协商。

4.32　set ixf1104 padding

该命令用于设置 ixf1104 自动填充开关。

命令格式

set ixf1104 padding {*type*}

参数说明

type　　自动填充开关，范围：enable / disable。

具体描述

set ixf1104 padding enable

说明：允许 ixf1104 自动填充。

4.33　set ixf1104 crc-appending

该命令用于设置 ixf1104 的 crc 校验开关。

命令格式

set ixf1104 crc-appending {*type*}

参数说明

type　　crc 校验开关，范围：enable / disable。

具体描述

set ixf1104 crc-appending enable

说明：允许 ixf1104 的 crc 校验。

4.34　show statistics

该命令用于查看端口号的统计信息。

命令格式

show statistics {*type*} {*data*}

参数说明

type 范围：pt（协议类型）/ sp（源端口）/ dp（目的端口）。
data 十进制数，范围：[0~65535]。

具体描述

show statistics sp 4000

说明：显示源端口号 4000 的统计信息。

4.35 show pmrc statistics

该命令用于查看 pmrc 的统计信息。

命令格式

show pmrc statistics

参数说明

无。

具体描述

show pmrc statistics

说明：显示 pmrc 的统计信息。结果如图所示：

```
GEPCI>show pmrc statistics
[RPL3]  Parity Check Error:  0
channel 0 Received   Error Packets :0  Long Packets :0  Short Packets :0
          Received   Packets Counter :172285565  Bytes Counter :243436612
          Received   Packets rate :238503(pps)  Bytes rate :119251700(Bps)
channel 1 Received   Error Packets :0  Long Packets :0  Short Packets :0
          Received   Packets Counter :0  Bytes Counter :0
          Received   Packets rate :0(pps)  Bytes rate :0(Bps)
channel 2 Received   Error Packets :0  Long Packets :0  Short Packets :0
          Received   Packets Counter :109560240  Bytes Counter :1748846880
          Received   Packets rate :151669(pps)  Bytes rate :121335680(Bps)
channel 3 Received   Error Packets :0  Long Packets :0  Short Packets :0
          Received   Packets Counter :0  Bytes Counter :0
          Received   Packets rate :0(pps)  Bytes rate :0(Bps)
 [PER] Received Packet :40986  Sent Packet :40988
 [TP] Received packet  from MUX :40988  to PD :50364
 [PD] Received Packet :50365  to IXF_IF :0  to MEM_IF :57697
 [CAM Interface] Received search keys from MUX. :34  to TP :34
 [MI]  Transmitted Packets  :4084525439  Frames  :3890934859  Useful Frames :533
582850  to PLXIF
 [IXF_IF] Sent Server Packet :0  Byte :0
        Sent Channel 0 Packet  Received   Packets :0  Bytes  :0
        Sent Channel 1 Packet  Received   Packets :0  Bytes  :0
        Sent Channel 2 Packet  Received   Packets :0  Bytes  :0
        Sent Channel 3 Packet  Received   Packets :0  Bytes  :0
```

4.36 show pmrc config

该命令用于查看 pmrc 的配置信息。

命令格式

show pmrc config

参数说明

无。

具体描述

show pmrc config

说明：显示 pmrc 的配置信息。结果如图所示：

```
GEPCI>show pmrc config
In function f_get_pmrc_config.
rx_parity EVEN
long_pkt=1540
short_pkt=40
rx 1 enable
rx 2 enable
rx 3 enable
rx 4 enable
remove VLAN
remove MPLS
```

4.37 set card

该命令用于选择设备。

命令格式

set card {*num*}

参数说明

num 卡号，范围：1 到使用的卡数。

具体描述

set card 1

说明：选择 1 号卡。

第5章 应用编程接口

5.1 接口概述

下表列出各调用接口的名称以及功能的概述,在其后部分将对每一接口及其功能进行详细的描述：

函数名	功能概述
pcap_open_live	获取采集句柄。
pcap_compile	解析规则和配置规则。
pcap_set_default_rule	配置设备的默认规则。
pcap_loop	开始进行采集.该函数内部处于循环状态。
pcap_breakloop	中断采集。
pcap_close	销毁采集句柄，释放资源。
pcap_stats_gfilter	获取统计信息。
pcap_version	库版本号。

5.2 基本数据结构

5.2.1 pcap_t

pcap_t 为设备句柄,用户使用该句柄访问设备和采集数据包。

5.2.2 pcap_handler

pcap_handler 为回调函数的指针.对采集到的数据包进行处理.该类型的声明如下:

```
typedef void (*pcap_handler)(u_char *usr_data, const struct pcap_pkthdr *hdr,
                const u_char *p);
```

其中,usr_data 为用户自定义数据；hdr 为数据包信息，p 为数据包指针，后两项是系统传入的参数，用户直接使用即可。

5.2.3 pcap_pkthdr

pcap_pkthdr 包含数据包信息，目前只支持数据封包的长度。

```
struct pcap_pkthdr {
                struct timeval ts;     /* time stamp */
                bpf_u_int32 caplen;        /* length of portion present */
                bpf_u_int32 len;     /* length this packet (off wire) */
                };
```

5.3　功能详述

5.3.1　pcap_open_live

通过该函数获取采集句柄. 应用程序通过采集句柄访问设备。

函数原型

pcap_t * pcap_open_live(const char *device, int snaplen, int promisc, int to_ms, char *errbuf)

入口参数

device　用于描述要打开设备的名称,如果该参数为 NULL,则表明使用默认的设备. 一般情况下采用默认值。当该参数为 ethN, N 取大等于零的整数时，该 libpcap 包为标准 libpcap 包。

snaplen　采集数据封包的最大长度.即如果封包实际含有大于 snaplen 的长度,但也只采集该封包的前 snaplen 个字节，在本版本中不关心此参数。

promisc　描述本应用所需要的缓存空间大小，以 2M 字节为基本单位。一般情况下采用参数 1（表示需要分配 2M 字节空间）。

to_ms　超时参数，在本版本中不关心此参数。

出口参数

errbuf　如果该参数不为 NULL,则 errbuf 指向的内存空间一定能够容纳 GCAP_ERROR_BUFFER_SIZE 个字符,使用该口进行过滤规则过程中发生错误时,该接口会将发生错误的描述信息放入该指针指向空间,如果该参数为 NULL,则表明用户并不在意错误描述信息。

返回值

创建成功,返回设备句柄,否则返回 NULL,如果 errbuf 不为 NULL,则通过 errbuf 传递错误信息。

5.3.2　pcap_compile

该函数用于解析规则和配置规则到设备。

函数原型

int pcap_compile (pcap_t *handle, struct bpf_program *program, char *buf, int optimize, bpf_u_int32 mask)

入口参数

handle　通过 pcap_open_live 获得的采集句柄.用该句柄访问设备。该参数无默认值。

program　在本版本中不关心此参数,留扩展用。

buf　保存需要设置的过滤规则。规则的关键字用空格分开，比如 buf = "tcp port 80" 此规则解析为过滤 tcp 类型的数据包并且数据包的目的或者源端口为 80。

optimize　在本版本中不关心此参数,留扩展用。

mask　在本版本中不关心此参数,留扩展用。

返回值

成功返回 0,否则返回-1,错误原因可能是用户向该函数传递了非法的参数。

5.3.3 pcap_set_default_rule

该函数用于配置设备的默认规则。设备有一条默认规则用来处理不匹配任何规则的数据流。丢弃和转发是默认的两种处理。

函数原型

int pcap_set_default_rule(pcap_t * handle, char *act_str);

入口参数

handle	通过 pcap_open_live 获得的采集句柄.用句柄访问设备。该参数无默认值。
act_str	保存需要设置的默认过滤规则。丢弃和转发是默认的两种处理，比如 act_str = "drop" 丢弃不匹配任何规则的数据流; act_str = "forward" 转发不匹配任何规则的数据流。

返回值

成功返回 0,否则返回-1,错误原因可能是用户向该函数传递了非法的参数。

5.3.4 pcap_loop

开始进行采集.该函数内部处于循环状态,直到用户规定的 cnt 个数据封包为止.该函数每接收到一个数据封包,都会调用 callback 注册函数对数据包处理. 用户可调用下面描述的 gcap_breakloop 结束循环状态。

函数原型

int pcap_loop (pcap_t *handle, int cnt, pcap_handler callback, u_char *user) ;

入口参数

handle	通过 pcap_open_live 获得的设备句柄.用句柄访问设备。该参数无默认值。
cnt	限定接收数据包的个数，当接收满 cnt 个数据后 pcap_loop 返回。当 cnt 小于等于零时表示无限循环接收收据包。当没有数据包时此函数返回。
callback	注册的回调函数，该函数每接收到一个数据封包,都会调用 callback 注册函数对数据包处理。
user	用户自定义数据，传递给 callback 使用。

返回值

成功返回 0 ,否则返回-1,错误原因可能是用户传递了错误的参数。

5.3.5 pcap_breakloop

停止在指定数据采集句柄上的所有数据采集处理.如果在一线程中对某采集句柄进行了 pcap_loop 操作,可以在另外的线程中调用 pcap_breakloop 来中止该循环。

函数原型

void pcap_breakloop (pcap_t *handle) ;

入口参数

handle 通过 pcap_open_live 获得的设备句柄.用句柄访问设备。该参数无默认值。

5.3.6 pcap_close

销毁采集句柄.采集工作完成后,使用该函数释放采集句柄本身所使用的资源。

函数原型

int pcap_close(pcap_t * handle) ;

入口参数

handle 待销毁的采集句柄。

返回值

销毁成功返回 0, 否则返回-1, 失败的原因可能是该句柄仍然处于使用中, 或者是用户向该函数传递了非法的参数。

5.3.7 pcap_stats_gfilter

获取对应采集描述句柄的采集信息.

函数原型

static int pcap_stats_gfilter (pcap_t *handle, struct pcap_stat *stats);

入口参数

handle 通过 pcap_open_live 获得的采集句柄.用句柄访问设备。该参数无默认值。

出口参数

stats pcap 对应的采集句柄的统计信息.包括接收到和丢弃的数据包统计信息。

返回值

销毁成功返回 0, 否则返回-1。

5.3.8 pcap_version

获取 libpcap 库的版本号。

函数原型

uint32_t pcap_version() ;

返回值

libpcap 库的版本号,高 16 位返回主版本号,低 16 位返回次版本号。

5.4　规则配置

5.4.1　基本规则

规则是过滤网络数据包的根据，本系统从网络上过滤数据包，检查网络上数据包是否匹配设置的规则，假如匹配则接受或者按比率采集数据；否则就丢弃数据包。

基本规则是不可再分的规则，本系统中的基本规则如下表所示，每一种编号代表一种基本规则：

①	**src port** value
②	**dst port** value
③	**src host** value (如 1.1.1.1)
④	**dst host** value (如 2.2.2.2)
⑤	**src net** value (如 1.1.0.0)
⑥	**dst net** value (如 2.2.0.0)
⑦	**tcp**
⑧	**udp**
	icmp
	igmp
	pt value

5.4.2　规则的组合运算

该版本 libgcap 支持两种规则运算：“与”和“或”运算。

“与”运算用”and”表示，比如用“src port 80 and dst port 8080”表示基本规则①与上基本规则②形成一条新的规则。

“或”运算用”or”表示，比如用“src port 80 or dst port 8080”表示基本规则①或上基本规则②形成一条新的规则。有些关键字可以用来简化“或”运算，如下表所示：

port value	**dst port** value **or src port** value
net value	**dst net** value **or src net** value
host value	**dst host** value **or src host** value

按照从左至右的顺序解析规则，“与” 和“或”具有同等的有先级别。比如

规则 “src port 80 and dst port 8080 or net 192.168.88.1”可以解析为以下规则的组合：

“src port 80 and dst port 8080”

“src net 192.168.88.1”

“dst net 192.168.88.1”

5.4.3 过滤规则

过滤规则是规则加上动作构成的，是配置到设备中的规则。

Filter rule = rule + action,不加 action 则默认为 forward。 本系统目前支持的 action 有以下几种：

Index	Action
①	**Drop**
②	**Forward**
③	**Sample10** 和 **Sample16**

Drop：代表符合该规则的数据包将会被丢弃。

Forward：代表符合该规则的所有数据包将会被送往服务器。

Sample：代表符合该规则的数据包只有一部分可以送往服务器，另外一部分会被丢弃，但是会保证 session 的完整性。进行 sample 时，会将数据包的源 ip，目的 ip，源端口和目的端口通过 hash 运算算出一个 0－15 的值，然后以这个值去索引一个 16bit 的数中的某一位，当这位为 1 时，将数据包送往服务器，否则将这个数据包丢弃。对于 sample，提供了两种方式，Sample10_后面直接接一个 0－15 的数值，表示 16 分之几的 session 会被送往服务器。Sample16_后面直接接一个 16bit 的数。Sample10_5 等价于 sample16_001F。

5.5 接口函数使用举例

5.5.1 举例一……接收 tcp port 80 数据包

```
#include <pcap.h>
#include <stdio.h>

void pcap_print(u_char *user, const struct pcap_pkthdr *h, const
u_char *p)
{
    int i;

    for ( i = 0; i <h->len; i++)
    {
             printf( "%02x-", p[i] );
    }
    printf("\n");
}
int main()
{
   pcap_t *handle; /* the handle */
   char *dev=NULL; /* the device */
   char errbuf[PCAP_ERRBUF_SIZE]; /* err buf */
   struct bpf_program filter; /* filter, no used */
   char filter_app[] = "tcp port 80"; /* the rule */
   bpf_u_int32 mask; /* no used */
   bpf_u_int32 net; /* no used */
   struct pcap_pkthdr header; /* header pointer */
   const u_char *packet;

   /* Define the device */
   handle = pcap_open_live(dev, 1600, 1, 0, errbuf);
   pcap_compile(handle, &filter, filter_app, 0, net);
   pcap_setfilter(handle, &filter);
   pcap_loop(handle, -1, pcap_print, NULL); //this function
                                //to test the pcap_loop
   pcap_close(handle);
   return(0);
}
```

第6章　问与答

- **问：**安装好 GEPCI 千兆过滤卡后，在 windows 中，系统检测不到千兆过滤卡；或者在 linux 中插入驱动会出错。

  答：将 GEPCI 千兆卡从服务器中取出，用橡皮或餐巾纸轻轻擦拭千兆过滤卡的金手指部分。
- **问：**安装好 GEPCI 千兆过滤卡后，并且驱动安装正常，如何判断与交换机的连接是否建立起来。

  答：首先看千兆过滤卡侧面对应口的状态灯是否是亮的，然后运行 parse 程序，输入 show ixf1104 status 命令查看对应口的状态，只要 SerDes signal detect 项为 1 就代表连接建立起来了。
- **问：**安装好 GEPCI 千兆过滤卡后，驱动安装正常，并且对应口的状态灯是亮的，但是千兆过滤卡采不到数据包。

  答：修改一下千兆过滤卡的 auto-negotiation 参数，具体修改文件在 gepci_card_driver/device/ixf1104/ixf1104_app.h 中，查找 auto negotiation，然后修改相应参数。
- **问：**如何查看每个端口输入的数据包个数，字节数，以及输入的数据包速率（pps）和字节速率（Bps）。

  答：运行 parse 程序，输入 show pmrc statistics 命令，即可查看每个端口输入的数据包情况。
- **问：**千兆过滤卡是否支持单模光纤输入的方式。

  答：千兆过滤卡采用 SFP 接口方式，将光模块换成单模光模块即可支持单模光纤输入。
- **问：**千兆过滤卡提供的 libpcap 库是否支持标准的网卡的 libpcap 库。

  答：是支持的，在条用 pcap_open_live 打开设备时，输入的设备名为 eth 开头时，将调用标准的网卡 libpcap 库，要使用千兆过滤卡，输入的设备名应为 gepci。